



Received on 17 December 2018; received in revised form, 14 March 2019; accepted, 28 March 2019; published 01 September 2019

## REALISTIC DEFORMATION AND REMOVAL OF SOFT TISSUES MODELING FOR THE SIMULATION OF VIRTUAL SURGERY

K. Jayasudha <sup>\*1</sup> and Mohan G. Kabadi <sup>2</sup>

Visvesvaraya Technological University <sup>1</sup>, Belgaum - 590018, Karnataka, India.

Presidency University <sup>2</sup>, Bangalore - 560064, Karnataka, India.

### Keywords:

Collision detection, Deformation,  
Skin, Simulation, Soft tissue

### Correspondence to Author:

**K. Jayasudha**

#765/17, 6<sup>th</sup> Cross, Triveni Road,  
Diwanarapalya, Bangalore - 560054,  
Karnataka, India.

**E-mail:** jayanataraja@gmail.com

**ABSTRACT:** Most of the simulation methods for soft tissue modeling involve tetrahedral meshes which is quite complex and takes much computation time. Instead, this work attempts to make use of delaunay triangulated mesh that consists of unique mathematical properties suited for simulating soft tissues. Although, triangulated mesh is not so complex yet effective in producing elements of good quality. It even reduces computation time compared to the tetrahedral mesh by providing more geometric flexibility. In virtual surgery, it is essential to model the layers of soft tissues of human skin to perform a simulation of deformation and removal of cells. Based on this the multilayered model of skin prototype is developed in a pre-process and used for interactive modeling. This work presents a simple method for performing real-time collision detection in a virtual surgery environment. Also shows the efficient computation of collisions between the scalpel and delaunay triangulated mesh using a local collision detection function. The framework incorporates qualitative results obtained towards the simulation of surgical deformation and removal of soft tissues using appropriate algorithms. It also uses real-time texture mapping to enhance the visual realism.

**INTRODUCTION:** Virtual surgery methods involve simulating the surgery procedure using human soft tissues on the computer monitor. So that trainee surgeons not only have a look and feel experience but also reduces the cost of actual operation to be carried out by committing fewer mistakes. The process of deformation in soft tissue modeling always remained a bottleneck in many applications. Now a day's realistic simulation is becoming more popular as they considered being highly accurate. During the surgery simulation deformation occurs whenever the scalpel intersects across the soft tissues of the skin.

Hence, necessary algorithms are used to demonstrate the same. This work attempts to generate a 3D volumetric mesh of soft tissues, and then deform these meshes using deformation algorithm and also remove those intersecting cells using removal algorithm. All these are done in real time as accurate as possible with the required computation time.

To begin with, reviews of techniques are briefed for soft tissue modeling, collision detection, surgical instrument (scalpel) for interactive cutting, volumetric deformation, and real-time simulations. The detailed methodology and experimental results were shown at the end. A variety of works have been carried out towards modeling and simulation of deformation of soft tissues in applications of virtual surgery. The related works have been broadly classified in the areas of soft tissue modeling, collision detection, surgical instrument, volumetric deformation, and real-time simulations.

<p><b>QUICK RESPONSE CODE</b></p> 	<p><b>DOI:</b> 10.13040/IJPSR.0975-8232.10(9).4270-79</p> <hr/> <p>The article can be accessed online on <a href="http://www.ijpsr.com">www.ijpsr.com</a></p> <hr/> <p>DOI link: <a href="http://dx.doi.org/10.13040/IJPSR.0975-8232.10(9).4270-79">http://dx.doi.org/10.13040/IJPSR.0975-8232.10(9).4270-79</a></p>
---	--

**Soft Tissue Modeling:** Soft tissue modeling is the process of simulation of soft tissues present in human skin. Many works have been done about soft tissue modeling. Surface-based soft tissues are modeled using Finite Element Method (FEM). It is one of the most popular methods with mathematical background used for simulating deformations of soft tissues. Guan *et al.*,<sup>1</sup> and Nebel *et al.*,<sup>2</sup> makes use of FEM for the simulation of soft tissues and also for stress-strain analysis. Crichton *et al.*,<sup>3</sup> shows viscoelastic properties in different layers of skin using nanoindentation technique. Bouten *et al.*,<sup>4</sup> explains tissue development of cardiovascular tissue engineering, *i.e.*, tissue development of arteries and heart valves. Visscher *et al.*,<sup>5</sup> proposes skin imaging which includes visual assessment, photographic image collection and early detection of tissue injury.

Author Varkey *et al.*,<sup>6</sup> discusses the skin substitutes need to be used to reestablish the protective barriers of skin when loss of skin is extensive. Author yuan *et al.*,<sup>7</sup> explains prototype simulator used for laparoscopic rectal cancer surgery training. It is illustrated using a hybrid model of tetrahedrons and multilayer triangular mesh using position based dynamics framework. Jinao *et al.*,<sup>8</sup> discusses nonlinear soft tissue deformation using energy propagation method based on poisson energy propagation. Its dynamics are governed in the combination of propagation of the mechanical load with nonrigid mechanical dynamics. Attila *et al.*,<sup>9</sup> debriefs about biological soft tissue modeling features in LS-DYNA software used in computational biomechanics. It mainly aims at simulating elastic and viscous behavior of biological soft tissues. Dinesh *et al.*,<sup>10</sup> studies the mechanical properties of the human body using fast and easy measurement system for mapping tissue properties.

Liang *et al.*,<sup>11</sup> emphasizes on tissue sticking during electrosurgical cutting on the porcine liver. Results have shown that they have strong bonding with the electrode. Attention is given to microstructure of sticking tissue and bonding with electrodes. Hosso *et al.*,<sup>12</sup> presents a simulation of surgical interventions using soft tissue modeling using FEM. The analysis is done using Lagrangian framework on virtual patients to show tool-tissue interaction. Zbigniew *et al.*,<sup>13</sup> explains the

simplification of technology allowing 3D flexible printed models of pulmonary collateral arteries could help in the surgery planning phase in case of cardio vascular disorders. Also insights the human body behavior. Andrea *et al.*,<sup>14</sup> explains soft tissue deformation of face modeling using smoothed finite element modeling (S-FEM). This overcomes shift behavior of standard finite element modeling (FEM). Jinao *et al.*,<sup>15</sup> explains soft tissue simulation in real time using chain mail method. This method improves computational efficiency for isotropic, nonisotropic and heterogeneous materials along with the relaxation behavior of soft tissues. The above-mentioned papers use various complex techniques like FEM, governing equations, continuum model, *etc.* The proposed method uses simple Delaunay triangulated cubic spring model for modeling soft tissues.

**Collision Detection:** Collision detection is the method for testing the interpenetration between a virtual deformable organ and a rigid virtual tool controlled by the user or interpenetration between any two virtual objects. This is an important step to achieve before deformation. A survey on collision detection always provides a comprehensive classification of different procedures adopted. The author Kockara *et al.*,<sup>16</sup> mentions a survey of collision detection into two phases namely broad-phase and narrow-phase. Broad-phase includes three algorithms namely: All-Pair-Test, sweep and prune and hierarchical hash table. Narrow-phase includes four algorithms namely: feature based, volume based, simplex-based and spatial data structure. Out of the two, narrow phase gives more detailed information.

Narrow phase uses linear programming, medial axis, I-collide algorithm, *etc.*, whereas the broad phase uses sphere trees, c-trees, AABB trees, *etc.* Bielser *et al.*,<sup>17</sup> explains the same using axis-aligned bounding box hierarchy. Basdogan *et al.*,<sup>18</sup>, Gregory *et al.*,<sup>19</sup> explain fast and accurate collision detection for haptic interaction. Basdogan *et al.*,<sup>18</sup> develops a training system to simulate laparoscopic procedures. Whereas, Gregory *et al.*,<sup>19</sup> uses H-COLLIDE, a framework that supports six degrees of freedom haptic devices for collision tests between pair of 3D objects and flexible surfaces. Author Benjones *et al.*,<sup>20</sup> explains

collision detection using example-based deformable bodies.

These are in turn union of spheres. These spheres are blend together with their position and radii almost similar to triangle mesh collisions. Fazioli *et al.*,<sup>21</sup> demonstrates collision detection between deformable organ (soft body) and surgical tool (rigid body) using nivent Falcon 3D that gives the position of the rigid body in space. It also gives force feedback through the haptic device. All these papers use complex algorithms, but the proposed method uses simple line data set function for collision detection in real time using the cell search algorithm.

**Surgical Instrument:** A surgical instrument also called as the scalpel is used for interactive mesh cutting in a virtual surgery environment. The surgical instrument plays a vital role in simulating cutting procedures. It depends on several factors such as sharpness of scalpel at which skin ruptures, the thickness of needle and force applied on scalpel for piercing the skin. Nienhuys *et al.*,<sup>22</sup> explains a method for producing cuts in triangulated surfaces. Nguyen *et al.*,<sup>23</sup> uses a prototype to validate new surgical instrument in which the user tests the virtual insertion of a rod in 1, 2 and 3 screws. Author Cheng *et al.*,<sup>24</sup> presents an interactive cutting simulation model for soft tissue. To incorporate physical and mechanical properties, nonlinearity and viscoelasticity equations are used. Alexey *et al.*,<sup>25</sup> explains the segmentation of robotic instruments which correctly determines instrument position. It is carried out using deep neural network architectures.

All these are applied in a linear simulation cutting system with haptic feedback. Each paper encompasses surgical instrument using either robotics, stereoscopic vision, CAD, *etc.* Christoph *et al.*,<sup>26</sup> discusses the estimation of force acting between tissue and instrument during deformation caused by instrument using optical coherence tomography. Bardia *et al.*,<sup>27</sup> discusses the mechanics of bevel tip needle insertion into soft tissues. The model is developed in LS-DYNA software using Lagrangian formulation for solid-fluid interactions. This paper uses simple scalpel made out of an assembly of 3D objects using a visualization tool kit.

**Volumetric Deformation:** Volumetric deformation is the process of resizing the intersected vertices of the mesh model such as topology modifications, stretching, skinning, and transformations. Meier *et al.*,<sup>28</sup> presents a survey with a comparison of deformable models. Peng *et al.*,<sup>29</sup> proposes a 3D node snapping algorithm to modify the surface topology of objects using the haptic device. Milliron *et al.*,<sup>30</sup> presents a frame work for mesh warp with a conceptual and mathematical foundation. Author D. P. Fankell *et al.*,<sup>31</sup> explains thermo-poromechanics (TPM) finite element models to study mass, momentum and energy balance occurring in the artery wall.

The model predicted final tissue temperature and water content simulations. Stamatia *et al.*,<sup>32</sup> proposes tissue deformation recovery by providing useful feedback on instrument forces exerted on the tissue. Nadzeri *et al.*,<sup>33</sup> focuses on soft tissue simulation in real time using new local deformation methodology. This method evaluates the deformation range using stress evaluation method based on elastic theory. Barbera *et al.*,<sup>34</sup> debriefs about inserting correct probe position during tissue deformation using finite element method around the contact area. This paper describes volumetric deformation using a simple algorithm such that when scalpel intersects the cube, its intersection points are known and vertices are resized.

**Real-Time Simulation:** Real-time simulation is the process of interaction taking place between modeled mesh and virtual scalpel in time or two modeled objects at the same time with probable outcome. Seiler *et al.*,<sup>35</sup> presents an adaptive octree based approach for interactive cutting of deformable objects. Author Junjun pan *et al.*,<sup>36</sup> describes real-time dissection approach to organ geometry consisting of inner metaballs and surface mesh with texture information. This is integrated into VR-based laparoscopic surgery simulator with a haptic interface.

The work describes a simple rendering algorithm for visualizing the changes occurring to the skin prototype to simulate realistic and real-time implementation. Neeladrishekar *et al.*,<sup>37</sup> explains the stress-strain behavior of muscles in real time called dynamics of human musculature. Also studies different material properties of muscles,

tendons, and bones. All these above-mentioned papers give us an idea about how tissue modeling can be done, how to detect collisions concerning a surgical instrument, also to carry out these simulations in real time.

Although this paper does not use haptic feedback system an attempt is made to prove the concept of soft tissue modeling and deformation of human skin using visualization technique particularly relevant to virtual surgery. The next section discusses the overall conceptual view to carrying out the real-time simulations.

**MATERIALS AND METHODS:** A framework is suggested to formulate three different layers of skin. The proposed framework is based on Delaunay triangulated mesh. The generated triangulated mesh is a multilayered structure depicting human skin and is aligned to 3D textured mapping as a pre-processing step<sup>38</sup>. **Fig. 1** depicts the data flow diagram of the simulation framework. Initially, the scalpel and soft tissue layers are modeled. When scalpel intersects the multilayered model in real time, various changes occur at each layer of the soft tissue model.

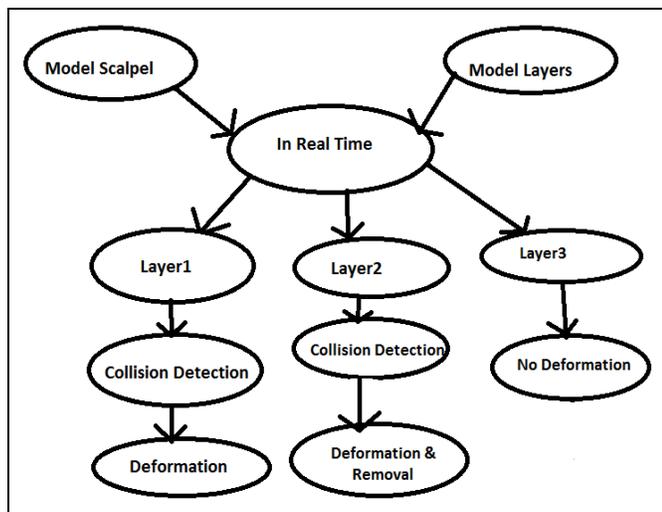


FIG. 1: DATA FLOW DIAGRAM

The collision detection procedures detect the interaction between the surgical tool (scalpel) and the tissue structures that can occur only at layer 1 and layer 2. This is because the first two layers are spongy and are deformable according to Nebel<sup>2</sup>, whereas the third layer is connected to muscles which are very hard and not movable by themselves. Therefore, it is assumed that muscles cannot be deformed by non-destructive external

forces. These interactions have to be computed. The next stage will be modeling incision where geometry updates have to be carried out enabling modification of triangulated mesh. This leads to the next stage called deformation. To compute deformation on the tissue structure, a deformation algorithm is run. Even to compute removal of intersected cells, a removal algorithm is run. Deformation cannot take place at the third layer as it leads to bone marrow, hence even though there is collision detection, there is no point of deformation taking place. The next section describes the methodology used. The soft tissue to be modeled is a three-layered structure which depicts the multiple layers of human skin. This is generated using triangulated mesh based on Delaunay criteria in a pre-process<sup>38</sup>. The Delaunay criterion states that “for n-dimensional cases, a circumsphere of each simplex within the mesh contains only the n+1 defining points of the simplex”. A nonlinear cubic spring mesh model<sup>39</sup> is studied for modeling soft tissues in real time simulation. Since the prototype is formulated using a Delaunay triangulated cube, each cube is composed of 18 edge-springs with linear spring constant A1 and cubic spring constant b1, and 18 diagonal-springs with linear spring constant A2 and cubic spring constant b2 is as shown in **Fig. 2**. The coefficients of linear terms A1 and A2 are based on equations of linear elasticity and equilibrium a condition as given below where E is Young's modulus, and v is Poisson's ratio.

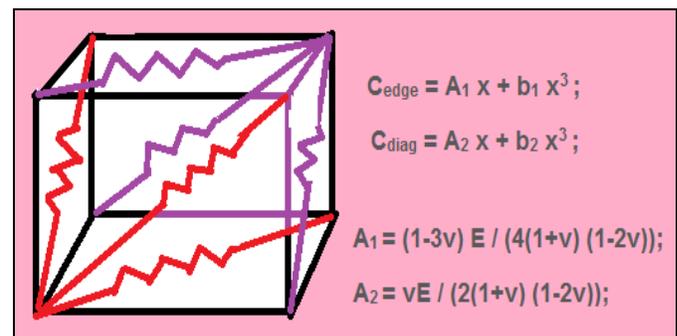


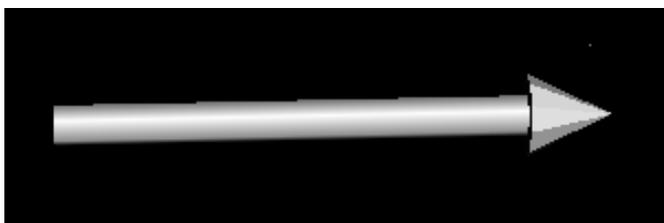
FIG. 2: CUBIC SPRING MODEL WITH EQUATIONS

According to Nebel<sup>2</sup>, human skin is composed of three layers: epidermis, dermis, and hypodermis/subcutaneous. The epidermis is the first layer that acts as a protective structure and is a little thin. The dermis is the second layer which is soft, thick and provides cushioning effect. Hypodermis/subcutaneous is the third layer which is thick and

closely connected to muscles. Author Varkey *et al.*,<sup>40</sup> describes the inner contents and thickness of each layer. According to this epidermis is about 50-100  $\mu\text{m}$  thick and is made up of two components namely stratum corneum (dead cell layer) and keratinizing epithelial cells (living epidermis).

The dermis is about 300-400  $\mu\text{m}$  thickness, also composed of two components called collagen and elastin fibers. According to Cerda and Mahadevan<sup>41</sup>, skin gets deformed only upon muscle contraction or outside mechanical force. As per their theory, the dermis is 10 times thicker than epidermis in terms of micrometers ( $\mu\text{m}$ ), *i.e.*  $(E_s / E_f) \approx 1000 \mu\text{m}$ , therefore  $\lambda \sim H_f$ . where  $\lambda$ : sinusoidal deflection profile of skin;  $H_f$ : thickness of dermis;  $E_s$ : elastic modulus of epidermis;  $E_f$ : elastic modulus of the dermis. That means when there is skin deflection, its effect is almost seen in the dermis layer because of its thickness. The mechanism of wrinkling of skin is beautifully explained by them through a simple understanding of many natural phenomenons like geometry, mechanics, physics, and biology.

Once the geometric model of skin prototype is ready, next step is to determine the line object coordinates in real-time. A line object is used as a preliminary virtual tool for detecting collision with the dataset. The line object is set with a viewable radius for easy interaction. The endpoints of the line object are determined in real time and checked if any part of the data set collides. A virtual scalpel is modeled by coupling the 3D objects from the available resources. The two objects such as cone and cylinder are combined in an assembly and then translated, rotated and scaled using visualization tool kit (VTK) to be rendered in the scene. The virtual scalpel model is shown in **Fig. 3**.



**FIG. 3: VIRTUAL SCALPEL**

The scalpel coordinates are then used to detect collision with the dataset. The next step is to simulate the true skin behavior, *i.e.* the operations that can be performed on the skin like collision

detection function which detects the intersection point of a scalpel with the skin prototype. Next is to implement deformation of cells such that whenever the scalpel is inserted into the skin, the cells get deformed or transformation of cells takes place, down the line if more pressure is applied to the scalpel it leads to skin cut, *i.e.* removal of cells at the later stage. At last, describes rendering function such that only those cells which are towards the camera are rendered. These algorithms are described as follows.

**Collision Detection Function:** The collision detection occurs when scalpel comes in contact with skin prototype. A line object is used as a preliminary virtual tool for detecting collision with the dataset. The line object is set with a viewable radius for easy interaction. The end points of the line object are determined in real time and checked if any part of the dataset has collided. The point along the line object that intersects with the dataset cell is stored in a 1D array of size 3, one each for x, y and z coordinates. It is an inbuilt function in VTK as shown in **Fig. 4**. It returns the intersection point (if any) and the cell which was intersected by finite line. It is a Boolean function which returns 0 (false) or 1 (true). The following function is run when the scalpel is brought in contact with the cube dataset. Here newlinep0 and newlinep1 are the extremes of intersecting finite line segment, t is the tolerance which is a parametric coordinate along the line segment ( a value between 0 and 1), x is an intersection point in data coordinates, pcoords is a parametric coordinates within the polygon, subId identifies a “subcell” within the cell. A surgical incision is done after achieving collision detection.

```
Is Collided = locator.IntersectWithLine (newlinep0, newlinep1, 0, t, x, pcoords, subId)
```

**FIG. 4: COLLISION DETECTION FUNCTION**

**Deformation Algorithm:** Cell deformation occurs either by changing its volume or shape of cells. After collision detection, the next step is the deformation stage which works according to the algorithm. Deformation algorithm is developed such that whenever the scalpel intersects the cube its intersection point is known, adjacent edges can be traced, and corresponding vertices are resized with a displacement of 1 or 2 units. The pseudo

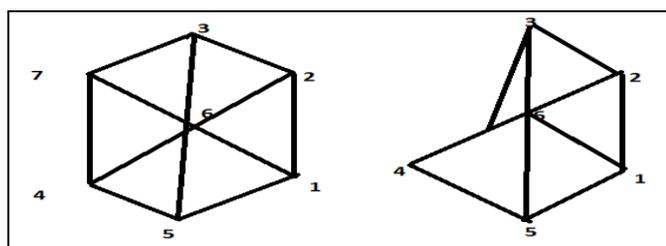
code for the deformation algorithm is as shown in **Fig. 5**. Here  $x$  is the point of the line of intersection. Each Delaunay cube contains 1.8 vertices, the distance from one vertex to other is 10. That means the distance from vertex 0 to vertex 1 is 10, the distance from vertex 0 to vertex 2 is 20, the distance from vertex 0 to vertex 3 is 30 and so on. So each vertex is taken as 10 mesh points; therefore the total control mesh points are 10..80. Here  $x[10..80]$  indicates the intersection points that ranges between 10 to 80 by control mesh points. As when the scalpel comes in contact with skin prototype, the intersection point is known. If intersection point ( $x$ ) is greater than or equal to 10, then cell vertices from the first position of cube are resized along  $x$ - $y$ - $z$ -axis and if intersection point ( $x$ ) is greater than or equal to 20, then cell vertices from the second position of cube are resized along  $x$ - $y$ - $z$ -axis and and if intersection point ( $x$ ) is greater than or equal to 30, then cell vertices from the third position of cube are resized along  $x$ - $y$ - $z$ -axis so on.

```

Deformation Of Cells(
for all vertex list [1..8] of Delaunay triangulated mesh do
    for all values of x[10..80] in terms of intersection points do
        if (x<=10)
            deform cells of epidermis layer by resizing control mesh points at position 1 along x-y-z axis
        else
            if ((x>10)&&(x<=20))
                deform cells of epidermis layer by resizing control mesh points at position 2 along x-y-z axis
            .....
        end for
    end for
end for
    
```

**FIG. 5: ALGORITHM FOR DEFORMATION OF CELLS.**

The same process is repeated up to 8 vertices of Delaunay cube to achieve realistic deformation of cells. **Fig. 6** clearly shows the pictorial representation of a cube consisting of 8 vertices along with deformation of the cell taking place at vertex 7. As the displacement increases, the deformation of skin also increases. Deformation occurs up to a certain level (threshold value), after which performs the removal algorithm.



**FIG. 6: DEFORMATION OF VERTEX 7 BEFORE AND AFTER**

**Removal Algorithm:** Removal algorithm is developed such that after resizing vertices, those cells are removed from the cube, to depict that if incision of the scalpel into the skin is continued, then skin gets cut. Those cells, which are in contact with scalpel, get cut, and removal of cells take place continuously till intersection point becomes zero. Cell removal is based on the logic that if more pressure is applied to the scalpel, the skin gets cut, *i.e.*, some cells are removed from skin prototype. The pseudo code for the removal algorithm is shown in **Fig. 7**. There will be 8 vertices in each cube and 80 intersection points. Removal of cells takes place in epidermis and dermis layers. This is because scalpel is piercing the skin model vertically touching both the layers.

When the point of intersection ( $x$ ) is greater than or equal to 10 then control mesh points at position1 are removed from epidermis and dermis layers of skin prototype along  $x$ - $y$ - $z$ -axis till intersection point becomes zero. If point of intersection ( $x$ ) is greater than or equal to 20, then control mesh points at position 2 are removed from epidermis and dermis layers of skin prototype along  $x$ - $y$ - $z$ -axis till intersection point becomes zero. Similarly, if the point of intersection ( $x$ ) is greater than or equal to 30 then control mesh points at position 3 are removed from epidermis and dermis layers of skin prototype along  $x$ - $y$ - $z$ -axis until intersection point turns to zero and so on. The same process is repeated up to 8 vertices of Delaunay cube to achieve realistic removal of cells.

```

Removal of Cells(
for all vertex list[1..8] of Delaunay triangulated mesh do
    for all values of x[10..80] in terms of intersection points do
        if (x<=10)
            remove cells of epidermis & dermis layers by removing control mesh points at position.1 along x-y-z axis
        else
            if ((x>10)&&(x<=20))
                remove cells of epidermis & dermis layers by removing control mesh points at position2 along x-y-z axis
            .....
        end for
    end for
end for
    
```

**FIG. 7: ALGORITHM FOR REMOVAL OF CELLS**

**Rendering Function:** Rendering function is developed for visualizing the cubes at periodic intervals like rendering the initial position of the cube, rendering the cube after it intersects with the scalpel, rendering after deformations occur, and rendering during removal of cells. This even renders, regaining the shape of the cube when the

scalpel is taken out *i.e.* when there is no intersection. Rendering of Delaunay triangulated mesh is achieved using VTK function “Render ( )”.

**RESULTS AND DISCUSSION:** The software used for the development of this application is in two folds. One is for visualization and the other one for developing the code. Visualization software is VTK 6.0 called Visualization Tool Kit and python is used for code development. Minimum hardware requirement is a pentium 4 processor with 4GB ram and 512MB graphics card. The following are the results obtained based on the algorithms explained above. The results were taken at each stage of the development of different algorithms. Whenever the scalpel intersects the cube data structure, the point

of intersection is known, its adjacent vertices can be traced, and corresponding vertices are deformed by setting control mesh points at that particular vertex. Then those cells are modified and updated only at the topmost layer, *i.e.* epidermis layer for visual appearance. **Fig. 8a** shows deformation of soft tissue prototype in a solid model and **Fig. 8b** shows the same result in the wireframe model. One can see the bifurcation of all the three layers that are texture mapped accordingly. **Fig. 9** shows the result of the deformation of cells from the cube data structure in an event action window. In the event action window, the modifications of time and array bounds are visible to show exactly at what point deformation of cells has occurred.

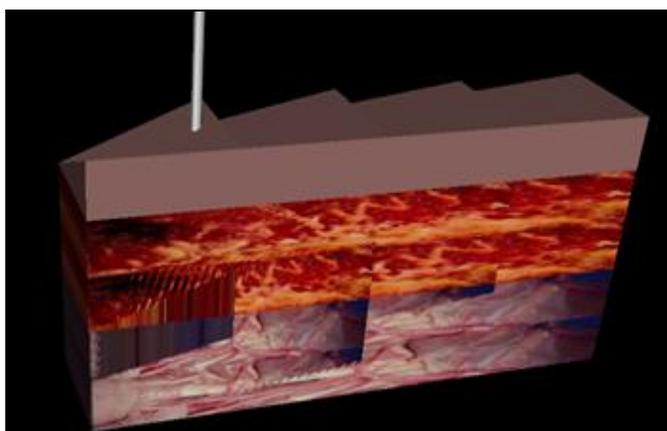


FIG. 8A: DEFORMATION PROCESS: SOLID MODEL

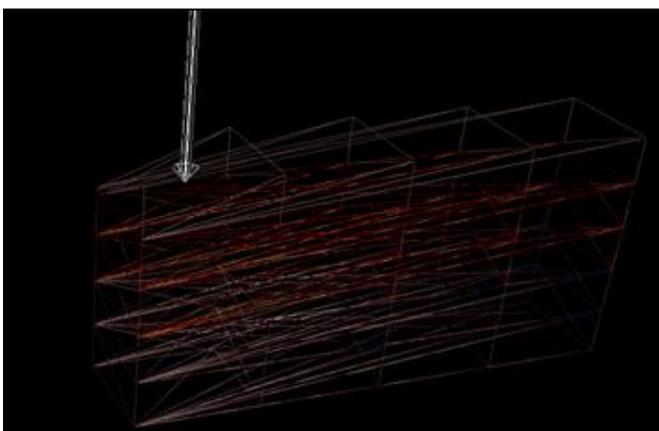


FIG. 8B: DEFORMATION PROCESS: WIREFRAME MODEL

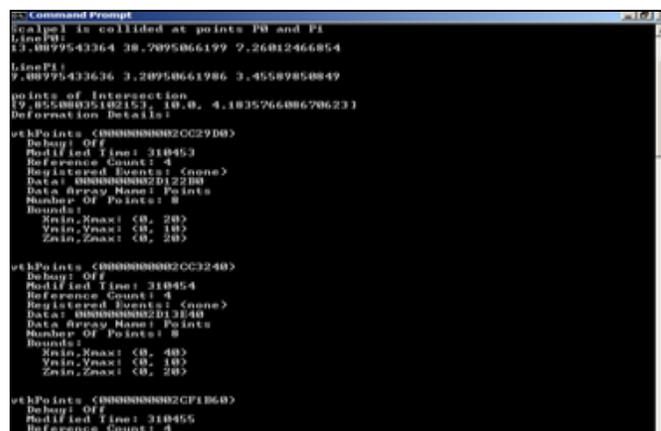


FIG. 9: EVENT ACTION WINDOW AFTER DEFORMATION

**Fig. 10a & b** shows the removal of soft tissue prototype cells in solid model and wireframe model. The solid model of **Fig. 10a** clearly shows the removal of cells, *i.e.* triangles in the epidermal layer. Also can observe slight deviation of cells, *i.e.* expansion of skin outwardly in the dermal layer.

This is to show that dermal layer also will have an impact when removal of cells taking place in the upper layer. The event action window is similar to **Fig. 9** which also shows the modified time and array bounds at which removal of cells has occurred.

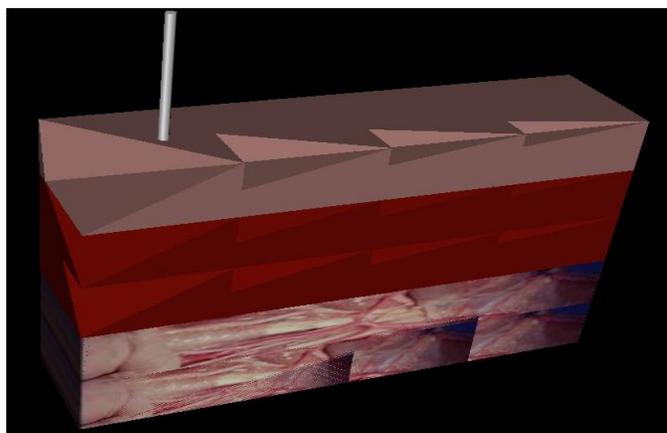


FIG. 10A: REMOVAL PROCESS: SOLID MODEL

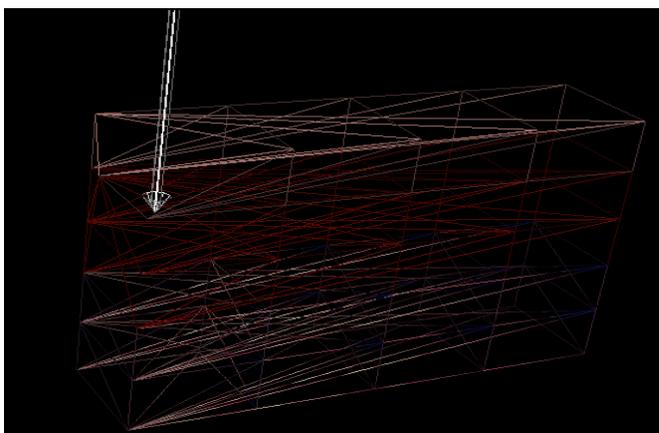


FIG. 10B: REMOVAL PROCESS: WIREFRAME MODEL

TABLE 1: RESULTS OF DEFORMATION AND REMOVAL PROCESS

Process name	Cell memory	Array Bounds			Execution time (sec)
		( $X_{min}$ , $X_{max}$ )	( $Y_{min}$ , $Y_{max}$ )	( $Z_{min}$ , $Z_{max}$ )	
Deformation at Epidermis	2CC9C80	(0,20)	(0,10)	(0,20)	0.139843
	2CCB810	(0,40)	(0,10)	(0,20)	0.139844
	2C7C6E0	(0,60)	(0,10)	(0,20)	0.139845
	2D2EB70	(0,80)	(0,10)	(0,20)	0.139846
	2DFC160	(0,20)	(-10,0)	(0,20)	0.139847
Removal at Dermis	2DFDCF0	(0,40)	(-10,0)	(0,20)	0.139848
	2CB72E0	(0,60)	(-10,0)	(0,20)	0.139849
	2DE3310	(0,80)	(-10,0)	(0,20)	0.139850
	2DEF7C0	(0,20)	(-20,0)	(0,20)	0.139851
	2F56FE0	(0,40)	(-20,0)	(0,20)	0.139852
	2F57D00	(0,60)	(-20,0)	(0,20)	0.139853
	2FBC840	(0,80)	(-20,0)	(0,20)	0.139854

Epidermal layer is texture mapped with skin color; dermal layer is shown as flesh like and sub-cutis layer with bone flesh like structures. Also to be noted about the thickness that each layer is shown. The outermost layer is usually thin and is represented as a single epidermal layer compared to the other two layers which are a little thicker and represented as double layers. The next process after deformation is the removal of cells. It is intended to show that if deformations occur for a longer period it leads to the removal of cells. During deformation stage, some vertices of the soft tissue prototype at the point of intersection gets updated. After updating the vertices, those cells are removed from the cube again by using point of intersection. The point at which the scalpel intersects the triangle, those triangles are removed at layer 1 and layer 2 by setting control mesh points at that particular position. Then these cells are modified and updated. This procedure is carried out for the first and second layers of skin prototype for visual appearance. The results of deformation and removal process are shown in the form of **Table 1**. It consists of the process name, the memory

location of cells (triangles), array bounds and execution time in seconds. These observations reveal that modeling and deformations on human soft tissues can be carried out even on the triangulated mesh.

**CONCLUSION:** In reality, actual operations require full involvement of the surgeon in the scene. Hence, it is important that complex scenes are to be displayed as real as possible in surgical applications. The attempt has been made to establish the logic of deformation and removal operations that can be performed on the multilayer model of skin prototype. The current work simulates the effect of real-time collision detection in a virtual surgery environment when scalpel comes in contact with triangulated mesh. Realistic scenes of skin prototype like deformation of cells, removal of cells are studied when the scalpel is being pierced into the skin. All these are implemented using deformation and removal algorithms. The system proves the concept of virtual surgery by providing realistic visualization of skin deformation. The following advancements

can be made in the future extending the scope and thus enhancing the capabilities of the present work. Enhancements can be made to include force feedback through haptic rendering, regaining the shape of skin when a scalpel is taken out and the concept of the flow of blood when skin gets cut.

**ACKNOWLEDGEMENT:** I am thankful to Mr. Brian Gee Chacko, Ibri College of technology, Oman, for his very valuable discussions on this topic. I also extend thanks to Dr. Shilpa Chaudri, Associate professor, MSRIT, Bangalore for the support and in augmenting many things.

**CONFLICT OF INTEREST:** The author declares no conflict of interest.

## REFERENCES:

- Guan, Qiu, Xiaochen Du, Yan Shao, Lili Lin, and Shengyong Chen. "Three-dimensional simulation of scalp soft tissue expansion using finite element method." *Computational and mathematical methods in medicine*, 2014.
- Nebel, Jean-Christophe. "Soft tissue modeling from 3D scanned data." In *Deformable avatars*. Springer US 2001: 85-97.
- Crichton, Michael L, Donose BC, Chen X, Raphael AP, Huang H, and Kendall MAF: The viscoelastic, hyperelastic and scale-dependent behavior of freshly excised individual skin layers. *Biomaterials* 2011; 32(20): 4670-81.
- Bouten CVC, Dankers PYW, Driessen-Mol A, Pedron S, Brizard AMA and Baaijens FPT: Substrates for cardiovascular tissue engineering." *Advanced Drug Delivery Reviews* 2011; 63(4): 221-41.
- Visscher MO: Imaging skin: past, present and future perspectives. *Giornale italiano di dermatologia e venereologia: organo ufficiale, Societ  italiana di dermatologia e sifilografia* 2010; 145(1): 11-27.
- Varkey, Mathew, Ding J and Edward E: Tredget. "Advances in skin substitutes-potential of tissue-engineered skin for facilitating anti-fibrotic healing. *Journal of Functional Biomaterials* 2015; 6(3): 547-63.
- Sui and Yuan: Real-time simulation of soft tissue deformation and electrocautery procedures in laparoscopic rectal cancer radical surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery* 2017; 13(4): e1827.
- Zhang and Jinao: Energy propagation modeling of nonlinear soft tissue deformation for surgical simulation. *Simulation* 2018; 94(1): 3-10.
- Nagy AP, Benson DJ, Kaul V and Palmer M: Constitutive Modeling of Biological Soft Tissues, *Biomedical* 2018, 15<sup>th</sup> International Conference on LS-DYNA users conference 2018; 1-6.
- Pai and Dinesh K: The human touch: measuring contact with real human soft tissues." *ACM Transactions on Graphics (TOG)* 2018; 37(4): 58.
- Zheng and Liang: Effect of high-frequency electric field on the tissue sticking of minimally invasive electrosurgical devices. *Royal Society Open Science* 2018; 5(7): 180125.
- El-Said, Ahmed S, Atta HMA and Hassanien AE: Interactive soft tissue modeling for virtual reality surgery simulation and planning. *International Journal of Computer Aided Engineering and Technology* 2017; 9(1): 38-61.
- Starosolski and Zbigniew: Soft tissue models: easy and inexpensive flexible 3D printing as help in surgical planning of cardiovascular disorders. *Medical Imaging 2017: Imaging Informatics for Healthcare, Research, and Applications*. Vol. 10138. International Society for Optics and Photonics, 2017.
- Mendizabal, Andrea, et al. "Face-based smoothed finite element method for real-time simulation of soft tissue." *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling*. Vol. 10135. International Society for Optics and Photonics, 2017.
- Zhang and Jinao: A new ChainMail approach for real-time soft tissue simulation. *Bioengineered* 2016; 7(4): 246-252.
- Kockara, Sinan, Tansel Halic, K. Iqbal, Coskun Bayrak, and Richard Rowe. "Collision detection: A survey." In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pp. 4046-4051. IEEE.
- Bielser, Daniel, and Markus H. Gross. "Interactive simulation of surgical cuts." In *Computer Graphics and Applications, 2000. Proceedings. The Eighth Pacific Conference on*, pp. 116-442. IEEE, 2000.
- Basdogan, Cagatay, Ho CH and Srinivasan MA: Virtual environments for medical training: Graphical and haptic simulation of laparoscopic common bile duct exploration. *IEEE/Asme Transactions on Mechatronics* 2001; 6(3): 269-85.
- Gregory, Arthur, Lin MC, Gottschalk S and Taylor R: A framework for fast and accurate collision detection for haptic interaction." In *ACM SIGGRAPH 2005*: 34.
- Jones and Ben: Efficient collision detection for example-based deformable bodies." *Proceedings of the Tenth International Conference on Motion in Games. ACM*, 2017.
- Fazioli, F: Implementation of a soft-rigid collision detection algorithm in an open-source engine for realistic surgical simulation. *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2016.
- Nienhuys, Wen H, and van der Stappen AF: A Delaunay approach to interactive cutting in triangulated surfaces. In *Algorithmic Foundations of Robotics V*. Springer Berlin Heidelberg, 2004: 113-29.
- Nguyen, Phan DM, Tonetti J and Thomann G: Evaluation of innovative surgical instrument using haptic interface in virtual reality. In *The second Vietnam Conference on Control and Automation*. 2013.
- Cheng and Qiang Q: An Interactive Meshless Cutting Model for Nonlinear Viscoelastic Soft Tissue in Surgical Simulators. *IEEE Access* 2017; 5: 16359-71.
- Shvets and Alexey A: Automatic instrument segmentation in robot-assisted surgery using deep learning. *2018 17<sup>th</sup> IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018.
- Christoph O: Towards force sensing based on instrument-tissue interaction. *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE, 2016.
- Kohn B and Hutapea P: A Real-Time Simulation for Path Planning in Needle Insertion Tasks, *ASME, 2015, Conference on Smart Materials, Adaptive Structures and Intelligent Systems* 2015.
- Meier, Ullrich, L pez O, Monserrat C, Juan MC and Alcaniz M: Real-time deformable models for surgery

- simulation: a survey. *Computer Methods and Programs in Biomedicine* 2005; 77(3): 183-197.
29. Peng, Jie, Ling L, and Squelch A: Hybrid surgery cutting using the snapping algorithm, volume deformation and haptic interaction. *J Man-Machine Technol* 2013; 2: 35-46.
  30. Tim M, Jensen RJ, Barzel R and Finkelstein A: A framework for geometric warps and deformations. *ACM Transactions on Graphics (TOG)* 2002; 21(1): 20-51.
  31. Fankell DP: A small deformation thermoporoelastic finite element model and its application to arterial tissue fusion. *Journal of Biomechanical Engineering* 2018; 140(3): 031007.
  32. Giannarou and Stamatia: Vision-based deformation recovery for intraoperative force estimation of tool-tissue interaction for neurosurgery. *Int Journal of Computer Assisted Radiology and Surgery* 2016; 11(6): 929-36.
  33. Omar N: Local deformation for Soft Tissue Simulation", *Bioengineered* 2016; 7(5): 291-297.
  34. Flack, Barbara, Makhinya M and Goksel O: Model-based compensation of tissue deformation during data acquisition for interpolative ultrasound simulation. 2016 IEEE 13<sup>th</sup> International Symposium on Biomedical Imaging (ISBI). IEEE 2016.
  35. Seiler, Martin, Steinemann D, Spillmann J and Harders M: Robust interactive cutting based on an adaptive octree simulation mesh. *The Visual Computer* 2011; 27(6): 519-29.
  36. Pan and Junjun: Real-time dissection of organs *via* hybrid coupling of geometric metaballs and physics-centric mesh-free method. *The Visual Computer* 2018; 34(1): 105-16.
  37. Kanjilal N and Chaudhuri P: Real-time Simulation of Human Musculature," 11<sup>th</sup> Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2018), December 18-22, 2018, Hyderabad, India.
  38. Jayasudha K, Kabadi MG and Chacko B: Multilayered Model of Soft Tissues for Surgery Simulation, *Integrated Intelligent Computing, Communication and Security (IICC&S), Studies in Computational Intelligence*, Springer, Singapore 2018; 771(2.7): 557-564.
  39. Shah, Rushina and Gupta A: Non-linear cubic spring-mesh model for simulating biological tissues. *Transactions of the Japanese Society for Medical and Biological Engineering. Supplement* 2013; 51: U-12.
  40. Varkey, Mathew, Ding J and Tredget EE: Advances in skin substitutes-potential of tissue-engineered skin for facilitating anti-fibrotic healing. *Journal of Functional Biomaterials* 2015; 6(3): 547-63.
  41. Cerda, Enrique and Mahadevan L: Geometry and physics of wrinkling. *Physical Review Letters* 2003; 90(7): 074302.

**How to cite this article:**

Jayasudha K and Kabadi MG: Realistic deformation and removal of soft tissues modeling for the simulation of virtual surgery. *Int J Pharm Sci & Res* 2019; 10(9): 4270-79. doi: 10.13040/IJPSR.0975-8232.10(9).4270-79.

All © 2013 are reserved by International Journal of Pharmaceutical Sciences and Research. This Journal licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

This article can be downloaded to **ANDROID OS** based mobile. Scan QR Code using Code/Bar Scanner from your mobile. (Scanners are available on Google Play store)